Concept Learning

Prof. Chia-Yu Lin

Yuan Ze University

2021 Spring

Thanks to the slides of Prof. Yu, Tian-Li from NTU.

Outline

- Learning From Examples

- Hypothesis

- Find-S

- Version space

- Candidate Elimination

- Inductive bias

Introduction

- Assume a given domain, e.g. objects, animals, etc.
- A concept can be seen as a subset of the domain, e.g. birds⊆animals
- → Extension of concept
- Task: acquire intensional concept description from training examples
- Generally we can't look at all objects in the domain

Learning a Class from Examples

- Training Examples for EnjoySport.

Sky	Temp	Humid	Wind	Water	Forecast	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Sunny	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

- What is the general concept?

Prototypical Concept Learning Task

Given:

- Instances X: Possible days, each described by the attributes Sky, $AirTemp$, $Humidity$, $Wind$, $Water$, $Forecast$.
- Target function c: $EnjoySport : X \rightarrow \{0, 1\}$
- Hypotheses H: Conjunctions of literals. E.g. $\langle ?, Cold, High, ?, ?, ? \rangle$.
- Training examples D: Positive and negative examples of the target function $\langle x_1, c(x_1) \rangle, \dots, \langle x_m, c(x_m) \rangle$

Determine:

- A hypothesis h in H such that $h(x) = c(x)$ for all x in D?
- A hypothesis h in H such that $h(x) = c(x)$ for all x in X?

Q1: What are instances?

- (A) Some training examples.
- (B) Some attributes or features.
- (C) Target or goal of machine learning.
- (D) None of the above.

Hypothesis

- Many possible representations.
- Here, h is conjunction of constraints on attributes.
- Each constraint can be
    - A specific value ($Water = Warm$).
    - Don't care ($Water = ?$).
    - May be empty ($Water = \phi$). $(\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$
- For example, $\langle Sky, AirTemp, Humid, Wind, Water, Forecast \rangle = \langle Sunny, ?, ?, Strong, ?, Same \rangle$.

這幾個都是用conjunction and 起來

The Inductive Learning Hypothesis

Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.

Instances and Hypothesis

$$x_1 = \langle Sunny, Warm, High, Strong, Cool, Same \rangle$$
$$x_2 = \langle Sunny, Warm, High, Light, Warm, Same \rangle$$

Instances X Hypotheses H
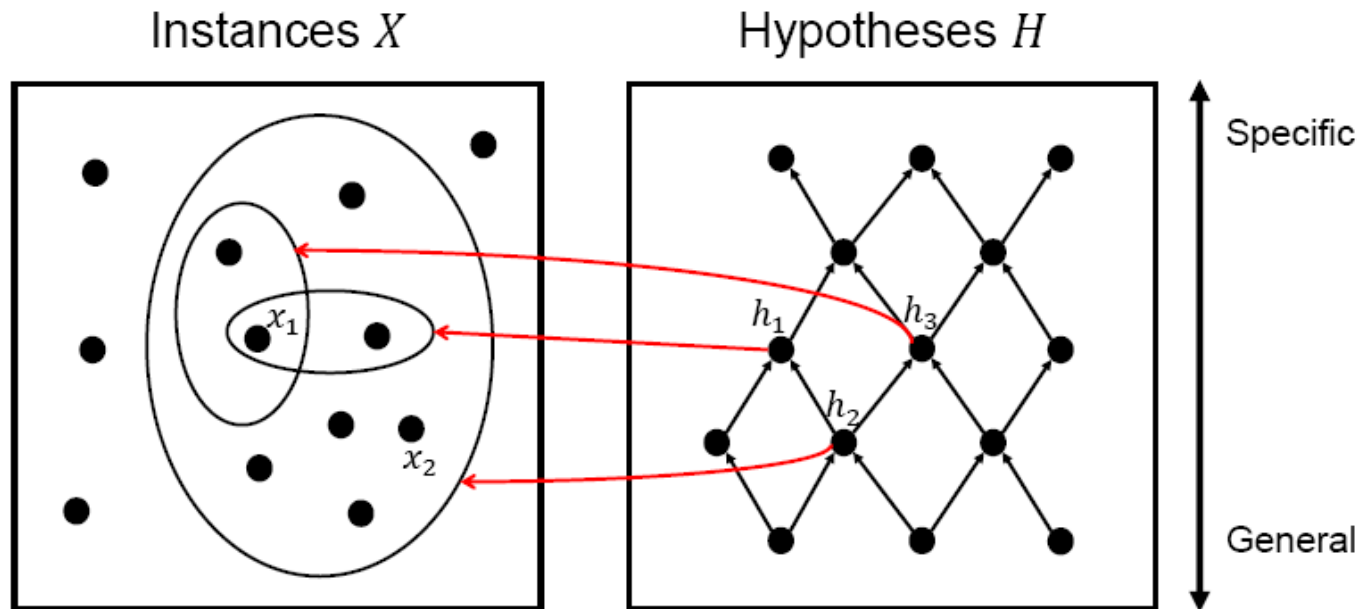
$$h_1 = \langle Sunny, ?, ?, Strong, ?, ? \rangle \quad x_1 \in m(h_1)$$
$$h_2 = \langle Sunny, ?, ?, ?, ?, ? \rangle \quad x_2 \notin$$
$$h_3 = \langle Sunny, ?, ?, ?, Cool, ? \rangle$$

The instance set of h_1 is included in h_2 => h_1 is more specific than h_2. or h_2 is more general than h_1.

Q2:

- In the case of the determination of Enjoysport or not, as described in the lecture, if h1=<Sunny,?,?,Light, Cool,?>, which of the following instances $x_1 \in m(h_1)$?

- (A) x_1=<Sunny, Warm, High, Strong, Cool, Same>
- (B) x_2=<Rainy, Cold, High, Light, Warm, Same>
- (C) x_3=<Sunny, Cold, High, Light, Cool, Change >
- (D) x_4=< Sunny, Warm, High, Strong, Cool, Change >

Models and "More General Than"

Definition: Model $m(h)$

Model $m(h)$ is a set of instances x where $h(x)$ is true:
$$m(h) = \{x \mid h(x) = true\}$$

Definitions

$h_1 <_g h_2$ iff $m(h_1) \subset m(h_2)$.
$h_1 \leq_g h_2$ iff $m(h_1) \subseteq m(h_2)$.

FIND-S Algorithm

FIND-S

1   Initialize h to the most specific hypothesis in H.
2   **for each** positive training instance x
3   **for each** attribute constraint a_i in h
4   If a_i in h is NOT satisfied by x, replace a_i in h by the next
5   more general constraint that is satisfied by x.
6   Output hypothesis h.

Hypothesis Space Search by Find-S

$$x_1 = \langle Sunny, Warm, Normal, Strong, Warm, Same \rangle : +$$
$$x_2 = \langle Sunny, Warm, High, Strong, Warm, Same \rangle : +$$
$$x_3 = \langle Rainy, Cold, High, Strong, Warm, Change \rangle : -$$
$$x_4 = \langle Sunny, Warm, High, Strong, Cool, Change \rangle : +$$
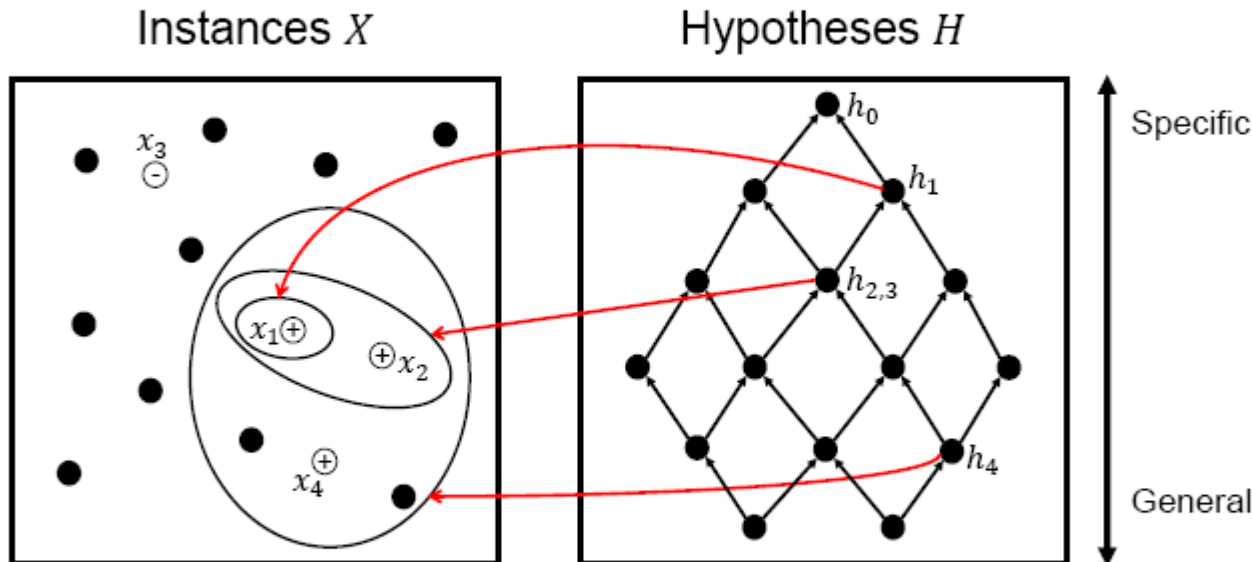
$$h_0 = \langle \phi, \phi, \phi, \phi, \phi, \phi \rangle$$
$$h_1 = \langle Sunny, Warm, Normal, Strong, Warm, Same \rangle$$
$$h_2 = \langle Sunny, Warm, ?, Strong, Warm, Same \rangle$$
$$h_3 = \langle Sunny, Warm, ?, Strong, Warm, Same \rangle$$
$$h_4 = \langle Sunny, Warm, ?, Strong, ?, ? \rangle$$

Properties of Find-S

- For hypothesis spaces that are described by conjunctions of attributes constraints, FIND-S is guaranteed to output the most specific hypothesis within the hypothesis space that is consistent with positive training examples.

- The output is also consistent with negative training examples provided the correct concept is in the hypothesis space.

Proof

$$c \in H \Rightarrow h \leq_g c \Rightarrow m(h) \subseteq m(c)$$
$$c(x) : - \Rightarrow x \notin m(c) \Rightarrow x \notin m(h) \Rightarrow h(x) : -$$

- Has the learner converged to the correct target concept?
- Why prefer the most specific hypothesis?
- Are the training examples consistent?
- What if there are several maximally specific consistent hypotheses?

Version Space (VS)

Definition: Consistency

A hypothesis h is **consistent** with a set of training examples D of target concept c if and only if $h(x) = c(x)$ for all $\langle x, c(x) \rangle$ in D.

$$Consistent(h, D) \equiv (\forall \langle x, c(x) \rangle \in D) \; h(x) = c(x)$$

Definition: Version Space

The version space, $VS_{H,D}$, with respect to hypothesis space H and training examples D, is the subset of hypotheses from H consistent with all training examples in D.

$$VS_{H,D} \equiv \{h \in H \mid Consistent(h, D)\}$$

Q3:

- What is the relation between the version space $VS_{H,D}$, the hypothesis H and the training example D?

- (A) $VS_{H,D}$ is a set of instances that are consistent with.
- (B) $VS_{H,D}$ is the subset of H.
- (C) $VS_{H,D}$ may contain some hypotheses that are not consistent with D.

List-Then-Eliminate Algorithm

LIST-THEN-ELIMINATE

1   VS = a set containing every hypothesis in H.
2   **for each** training example $\langle x, c(x) \rangle \in D$
3   Remove from VS any hypothesis h for which $h(x) \neq c(x)$.
4   Output the list of hypotheses in VS.

- LIST-THEN-ELIMINATE outputs all hypotheses that are consistent with examples.
- Theoretically, it works for any finite version spaces.
- However, the requirement of memory is impractical.
- We need more compact representation of VS.

Representing Version Spaces (1/2)

Definition: General Boundary

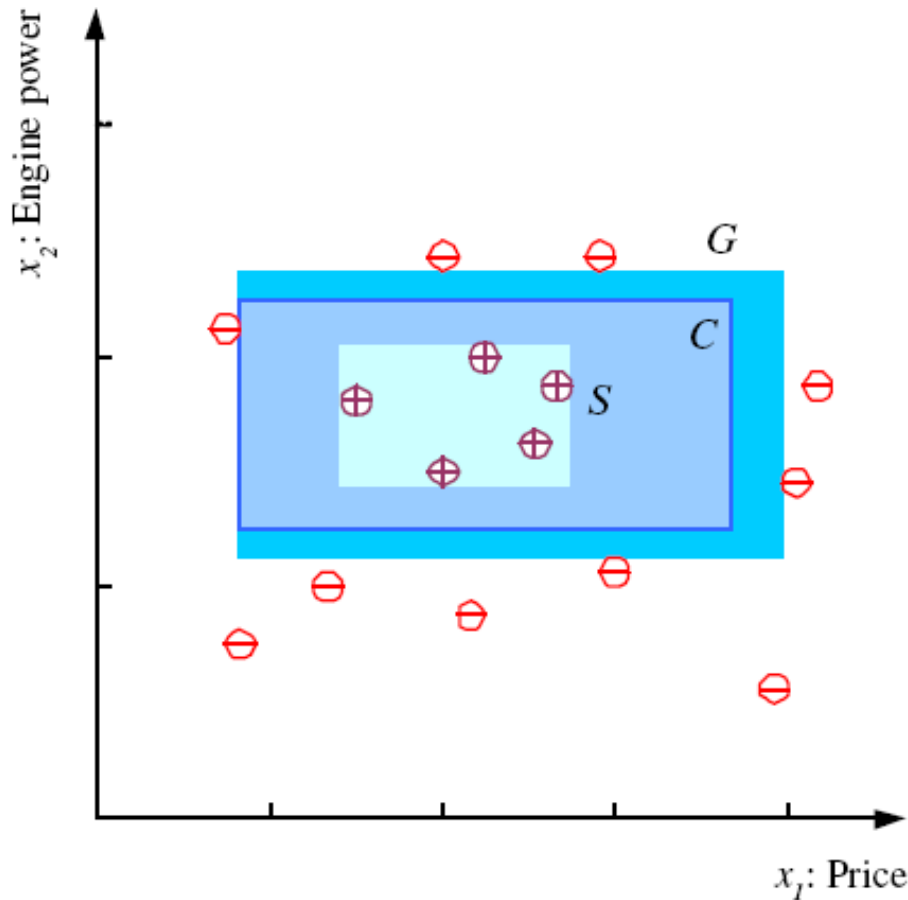The **general boundary**, G, of version space $VS_{H,D}$ is the set of its maximally general members.

$$G \equiv \{g \in H \,|\, Consistent(g, D) \wedge (\neg \exists g' \in H \,(g <_g g') \wedge Consistent(g', D))\}$$

Definition: Specific Boundary

The **specific boundary**, S, of version space $VS_{H,D}$ is the set of its maximally specific members.
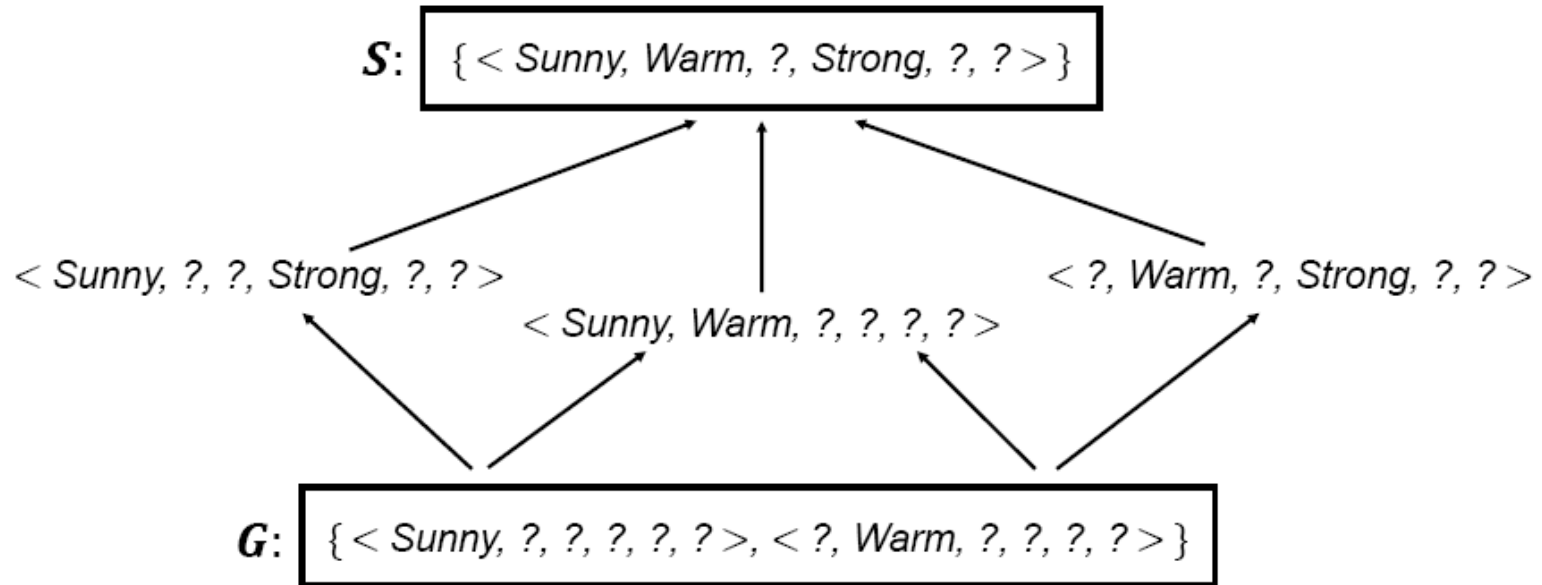
$$S \equiv \{s \in H \,|\, Consistent(s, D) \wedge (\neg \exists s' \in H \,(s' <_g s) \wedge Consistent(s', D))\}$$

Representing Version Spaces (2/2)

S, the most specific hypothesis, *G*, the most general hypothesis, and any $h \in$ H between *S* and *G* is consistent and make up the version space
(Mitchell, 1997)

Representing Version Spaces

S: $\{< Sunny, Warm, ?, Strong, ?, ?>\}$

$< Sunny, ?, ?, Strong, ?, ?>$ $< ?, Warm, ?, Strong, ?, ?>$

$< Sunny, Warm, ?, ?, ?, ?>$

G: $\{< Sunny, ?, ?, ?, ?, ?>, < ?, Warm, ?, ?, ?, ?>\}$

Version Space Representation Theorem

Every member of the version space lies between these two boundaries.

$$VS_{H,D} = \{h \in H \mid (\exists s \in S)(\exists g \in G)\; s \leq_g h \leq_g g\}.$$

Candidate-Elimination Algorithm

❶ G = set of maximally general hypotheses in H.
❷ S = set of maximally specific hypotheses in H.
❸ For each training example d, do
- If d is positive,
    - Remove from G any hypothesis <u>inconsistent</u> with d.
    - For each $s \in S$ <u>inconsistent</u> with d
        - Remove s from S.
        - Add to S all minimal generalization h of s s.t. h is consistent with d and some member of G is more general than h.
        - Remove from S any hypothesis that is more general than another hypothesis in S.

$h_1 <_g$ ✖

- If d is negative,
    - Remove from S any hypothesis inconsistent with d.
    - For each $g \in G$ inconsistent with d
        - Remove g from G.
        - Add to G all minimal specification h of g s.t. h is consistent with d and some member of S is more specific than h.
        - Remove from G any hypothesis that is more specific than another hypothesis in G.

Candidate Elimination Example (1/4)

- Training examples
  - $\langle Sunny, Warm, Normal, Strong, Warm, Same \rangle : +$
  - $\langle Sunny, Warm, High, Strong, Warm, Same \rangle : +$
  - $\langle Rainy, Cold, High, Strong, Warm, Change \rangle : -$
  - $\langle Sunny, Warm, High, Strong, Cool, Change \rangle : +$

- Initially,
  - $S_0 = \langle \phi, \phi, \phi, \phi, \phi, \phi, \phi \rangle$
  - $G_0 = \langle ?, ?, ?, ?, ?, ?, ? \rangle$

S_0: $\{< \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset >\}$

S_1:

S_2:

G_0, G_1, G_2: $\{< ?, ?, ?, ?, ?, ? >\}$

Training Examples:

S_2, S_3: $\{< Sunny, Warm, ?, Strong, Warm, Same >\}$

G_3:

G_2: $\{< ?, ?, ?, ?, ?, ? >\}$

Sunny? Light? Same?
Warm? Cold?

Training Example:
3. $< Rainy, Cold, High, Strong, Warm, Change >$, $EnjoySport = No$

Candidate Elimination Example (4/4)

S_3: $\{< Sunny, Warm, ?, Strong, Warm, Same >\}$

S_4: $\{< Sunny, Warm, ?, Strong, ?, ? >\}$

G_4: $\{< Sunny, ?, ?, ?, ?, ? > < ?, Warm, ?, ?, ?, ? >\}$

G_3: $\{< Sunny, ?, ?, ?, ?, ? > < ?, Warm, ?, ?, ?, ? > < ?, ?, ?, ?, ?, Same >\}$

Training Example:
4. $< Sunny, Warm, High, Strong, Cool, Change >$, EnjoySport = Yes

Properties of Candidate Elimination

- Does it converge to the correct concept?
  - Yes if (1) there is no error in the training examples and (2) some hypothesis in H correctly describes the target concept.
  - When one of these (or both) doesn't hold, given enough data, eventually S and G crosses, and yields an empty VS.
- What training example should the learner request next?
  - A good query should be classified as positive by some $h \in H$ and negative by others.
  - An optimal query should be half-half, then only $\lceil \lg |VS| \rceil$ such queries are needed to learn the exact concept if any.
- How can partially learned concepts be used?
  - An instance is positive for the target concept if it satisfies every member in S.
  - An instance is negative for the target concept if it satisfies none in G.
  - We may calculate confidence for other instances given some prior.

Inductive Bias

- We talked about the difficulty where the target concept is not in the hypothesis space.
- Why not using a hypothesis space which includes every possible hypothesis?
- How does $|H|$ affect the generalization of the learner?
- How does $|H|$ affect required number of training examples?

- Inductive bias of CANDIDATE-ELIMINATION:
  The target concept is contained in the hypothesis space H.

- Our previous conjunctive hypothesis space:
  (sunny, cloudy, rainy, don't care)   (all attribute is empty)
  - Contains only $4 \times 3 \times 3 \times 3 \times 3 \times 3 + 1 = 973$ concepts, very biased.
  - $\langle Sunny, Warm, Normal, Strong, Cool, Change \rangle : +$
    $\langle Cloudy, Warm, Normal, Strong, Cool, Change \rangle : +$
    $\langle Rainy, Warm, Normal, Strong, Cool, Change \rangle : -$
    Our algorithm will find zero hypothesis since it can't learn disjunctions
    such as $Sky = Sunny \vee Sky = Cloudy$.

    要包住sunny and cloudy => don't care => 但是又會包到rainy

- Consider an unbiased hypothesis space.
  - $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$ instances in total $\Rightarrow 2^{96} \simeq 10^{28}$ distinct target concepts. (instance不能有don't care. Hypothesis才能有don't care)
  - We never have to worry whether the target concept is in H.

$973 << 10^{28}$ =>We only consider very limited hypothesis =>Very bias

Futility of Bias-Free Learning

- No generalization in such hypothesis space!
    - Positive examples: x_1, x_2, x_3; negative: x_4, x_5.
    - $S = \{x_1 \vee x_2 \vee x_3\}$.
    - $G = \{\neg x_4 \wedge \neg x_5\}$.

- A learner that makes no prior assumptions regarding the identity of the target concept has no rational basis for classifying any unseen instances.

- No-free-lunch theorem.

- No bias, no learning. 要machine學 => 要給他一些preference!

Summary

- Concept learning can be cast as a problem of searching through a large predefined space of potential hypotheses.

- Search with the general-to-specific partial ordering.

- FIND-S search from specific to general and outputs the most specific consistent hypothesis.

- CANDIDATE-ELIMINATION keeps the most general (G) and specific (S) hypothesis. It shrinks VS during the search by relaxing S with positive examples and restricting G with negative ones.

- Inductive learning algorithms are able to classify unseen data because of implicit inductive bias.